

Refine Search

Search Results -

Terms	Documents
L7 and offset	18

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search:

L9

Search History

DATE: Monday, November 19, 2007 [Purge Queries](#) [Printable Copy](#) [Create Case](#)

Set Name Query

side by side

Hit Count Set Name

result set

DB=USPT; PLUR=NO; OP=OR

<u>L9</u>	L7 and offset	18	<u>L9</u>
<u>L8</u>	L7 and object	38	<u>L8</u>
<u>L7</u>	L5 and inlining	40	<u>L7</u>
<u>L6</u>	L5 and closure	18	<u>L6</u>
<u>L5</u>	L4 and virtual and class	531	<u>L5</u>
<u>L4</u>	(717/108 717/116 717/140 717/141 717/142 717/143 717/144).ccls.	1624	<u>L4</u>
<u>L3</u>	sweeney.ab.	5	<u>L3</u>
<u>L2</u>	L1 and sweeney.ab.	0	<u>L2</u>
<u>L1</u>	(virtual ADJ function) and offset	226	<u>L1</u>

END OF SEARCH HISTORY



USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

virtual function fixed offset

SEARCH


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used: virtual function fixed offset

Found 68,883 of 215,186

Sort results by

relevance

[Save results to a Binder](#)Try an [Advanced Search](#)Try this search in [The ACM Guide](#)

Display results

expanded form

[Search Tips](#)☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Computer security \(SEC\): Improving address space randomization with a dynamic offset randomization technique](#)



Haizhi Xu, Steve J. Chapin

April 2006 **Proceedings of the 2006 ACM symposium on Applied computing SAC '06**

Publisher: ACM Press

Full text available: [pdf\(176.95 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Address Space Randomization (ASR) techniques randomize process layout to prevent attackers from locating target functions. Prior ASR techniques have considered single-target attacks, which succeed if the attacker can locate a single, powerful system library function. These techniques are not sufficient to defend against *chained return-into-lib(c) attacks*, each of which calls a sequence of system library functions in order. In this paper, we propose a new ASR technique, *code islands*, ...

Keywords: address space randomization, code islands, denial-of-service attacks, derandomization attacks, intrusion mitigation, randomization



2 [Space and time-efficient memory layout for multiple inheritance](#)



Peter F. Sweeney, Joseph (Yossi) Gil

October 1999 **ACM SIGPLAN Notices , Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '99**, Volume 34 Issue 10

Publisher: ACM Press

Full text available: [pdf\(2.30 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Traditional implementations of multiple inheritance bring about not only an overhead in terms of run-time but also a significant increase in object space. For example, the number of compiler-generated fields in a certain object can be as large as quadratic in the number of its subobjects. The problem of efficient object layout is compounded by the need to support two different semantics of multiple inheritance: shared, in which a base class inherited along distinct ...

3 [Invited talk: Intermediate-representation recovery from low-level code](#)



Thomas Reps, Gogul Balakrishnan, Junghee Lim

January 2006 **Proceedings of the 2006 ACM SIGPLAN symposium on Partial evaluation and semantics-based program manipulation PEPM '06**

Publisher: ACM Press

Full text available:  [pdf\(301.12 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The goal of our work is to create tools that an analyst can use to understand the workings of COTS components, plugins, mobile code, and DLLs, as well as memory snapshots of worms and virus-infected code. This paper describes how static analysis provides techniques that can be used to recover intermediate representations that are similar to those that can be created for a program written in a high-level language.

4 Language-based security: SELF: a transparent security extension for ELF binaries



Daniel C. DuVarney, V. N. Venkatakrishnan, Sandeep Bhatkar

August 2003 **Proceedings of the 2003 workshop on New security paradigms NSPW '03**

Publisher: ACM Press

Full text available:  [pdf\(1.05 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

The ability to analyze and modify binaries is often very useful from a security viewpoint. Security operations one would like to perform on binaries include the ability to extract models of program behavior and insert inline reference monitors. Unfortunately, the existing manner in which binary code is packaged prevents even the simplest of analyses, such as distinguishing code from data, from succeeding 100 percent of the time. In this paper, we propose SELF, a security-enhanced ELF (Executable ...

5 A principled approach to operating system construction in Haskell



Thomas Hallgren, Mark P. Jones, Rebekah Leslie, Andrew Tolmach

September 2005 **ACM SIGPLAN Notices , Proceedings of the tenth ACM SIGPLAN international conference on Functional programming ICFP '05**, Volume 40 Issue 9

Publisher: ACM Press

Full text available:  [pdf\(154.82 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We describe a monadic interface to low-level hardware features that is a suitable basis for building operating systems in Haskell. The interface includes primitives for controlling memory management hardware, user-mode process execution, and low-level device I/O. The interface enforces memory safety in nearly all circumstances. Its behavior is specified in part by formal assertions written in a programming logic called P-Logic. The interface has been implemented on bare IA32 hardware using the G ...

Keywords: Haskell, hardware interface, monads, operating systems, programming logic, verification

6 Security: Segment protection for embedded systems using run-time checks



Matthew Simpson, Bhuvan Middha, Rajeev Barua

September 2005 **Proceedings of the 2005 international conference on Compilers, architectures and synthesis for embedded systems CASES '05**

Publisher: ACM Press

Full text available:  [pdf\(198.51 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The lack of virtual memory protection is a serious source of unreliability in many embedded systems. Without the segment-level protection it provides, these systems are subject to memory access violations, stemming from programmer error, whose results can be dangerous and catastrophic in safety-critical systems. The traditional method of testing embedded software before its deployment is an insufficient means of detecting and debugging all software errors, and the reliance on this practice is a ...

Keywords: MMU, MPU, compilers, embedded systems, memory safety, reliability, run-time checks, safe languages, segment protection, segmentation violations, virtual

memory

7 6-1 Visual perception & image processing: Space extension: the perceptual presence perspective



Jane Hwang, Gerard J. Kim, Albert Skip Rizzo

June 2004 **Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry VRCAI '04**

Publisher: ACM Press

Full text available: pdf(407.88 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The sense of presence has been the main goal of many virtual reality systems. Consequently, many research results have identified elements that contribute to high presence. However, there has been little work in applying such results to specific system and application design. In this paper, we present a model of presence that is based on the capabilities and evolutionary nature of the human perceptual system. We illustrate how we apply the model to configuring a particular virtual reality displa ...

Keywords: immersadesk, model, presence, spatial continuity

8 Exploiting perception in high-fidelity virtual environments: Exploiting perception in high-fidelity virtual environments



Additional presentations from the 24th course are available on the citation page

Mashhuda Glencross, Alan G. Chalmers, Ming C. Lin, Miguel A. Otaduy, Diego Gutierrez
July 2006 **ACM SIGGRAPH 2006 Courses SIGGRAPH '06**

Publisher: ACM Press

Full text available: pdf(5.07 MB) mov(68:6 MIN)

Additional Information: [full citation](#), [appendices and supplements](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

The objective of this course is to provide an introduction to the issues that must be considered when building high-fidelity 3D engaging shared virtual environments. The principles of human perception guide important development of algorithms and techniques in collaboration, graphical, auditory, and haptic rendering. We aim to show how human perception is exploited to achieve realism in high fidelity environments within the constraints of available finite computational resources. In this course w ...

Keywords: collaborative environments, haptics, high-fidelity rendering, human-computer interaction, multi-user, networked applications, perception, virtual reality

9 Research papers III: Catenation and specialization for Tcl virtual machine performance



Benjamin Vitale, Tarek S. Abdelrahman

June 2004 **Proceedings of the 2004 workshop on Interpreters, virtual machines and emulators IVME '04**

Publisher: ACM Press

Full text available: pdf(188.95 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We present techniques for eliminating dispatch overhead in a virtual machine interpreter using a lightweight just-in-time native-code compilation. In the context of the Tcl VM, we convert bytecodes to native Sparc code, by concatenating the native instructions used by the VM to implement each bytecode instruction. We thus eliminate the dispatch loop. Furthermore, immediate arguments of bytecode instructions are substituted into the

native code using runtime specialization. Native code output fro ...

Keywords: Tcl, bytecode interpreters, just-in-time compilation, virtual machines

10 The multics system: an examination of its structure

Elliott I. Organick
January 1972 Book

Publisher: MIT Press

Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

This volume provides an overview of the Multics system developed at M.I.T.--a time-shared, general purpose utility like system with third-generation software. The advantage that this new system has over its predecessors lies in its expanded capacity to manipulate and file information on several levels and to police and control access to data in its various files. On the invitation of M.I.T.'s Project MAC, Elliott Organick developed over a period of years an explanation of the workings, concep ...

11 Virtual videography

 Rachel Heck, Michael Wallick, Michael Gleicher
February 2007 **ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)**, Volume 3 Issue 1

Publisher: ACM Press

Full text available:  [pdf\(5.41 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Well-produced videos provide a convenient and effective way to archive lectures. In this article, we offer a new way to create lecture videos that retains many of the advantages of well-composed recordings, without the cost and intrusion of a video production crew. We present an automated system called *Virtual Videography* that employs the art of videography to mimic videographer-produced videos, while unobtrusively recording lectures. The system uses the data recorded by unattended video ...

Keywords: Automated camera management, attention modeling, computational cinematography, video production


12 Course 17: Spatial augmented reality: merging real and virtual worlds: Modern approaches to augmented reality

 **Video files associated with this course are available from the citation page**

Oliver Bimber, Ramesh Raskar

August 2007 **ACM SIGGRAPH 2007 courses SIGGRAPH '07**

Publisher: ACM Press

Full text available:  [pdf\(46.17 MB\)](#) Additional Information: [full citation](#), [appendices and supplements](#), [abstract](#), [references](#), [index terms](#)

This tutorial discusses the Spatial Augmented Reality (SAR) concept, its advantages and limitations. It will present examples of state-of-the-art display configurations, appropriate real-time rendering techniques, details about hardware and software implementations, and current areas of application. Specifically, it will describe techniques for optical combination using single/multiple spatially aligned mirror-beam splitters, image sources, transparent screens and optical holograms. Furthermo ...

13 Smalltalk-80: the language and its implementation

Adele Goldberg, David Robson
January 1983 Book


Publisher: Addison-Wesley Longman Publishing Co., Inc.

Full text available:  pdf(33.56 MB) Additional Information: [full citation](#), [abstract](#), [cited by](#), [index terms](#), [review](#)

From the Preface (See Front Matter for full Preface)

Advances in the design and production of computer hardware have brought many more people into direct contact with computers. Similar advances in the design and production of computer software are required in order that this increased contact be as rewarding as possible. The Smalltalk-80 system is a result of a decade of research into creating computer software that is appropriate for producing highly functional and interactive ...

14 The implementation and evaluation of dynamic code decompression using DISE

 Marc L. Corliss, E. Christopher Lewis, Amir Roth
February 2005 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 4
Issue 1

Publisher: ACM Press

Full text available:  pdf(1.23 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Code compression coupled with dynamic decompression is an important technique for both embedded and general-purpose microprocessors. *Postfetch decompression*, in which decompression is performed after the compressed instructions have been fetched, allows the instruction cache to store compressed code but requires a highly efficient decompression implementation. We propose implementing postfetch decompression using a new hardware facility called *dynamic instruction stream editing* (DI ...

Keywords: Code compression, DISE, code decompression, dynamic instruction stream editing, dynamic instrumentation

15 GPGPU: general purpose computation on graphics hardware


 David Luebke, Mark Harris, Jens Krüger, Tim Purcell, Naga Govindaraju, Ian Buck, Cliff Woolley, Aaron Lefohn
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**

Publisher: ACM Press

Full text available:  pdf(63.03 MB) Additional Information: [full citation](#), [abstract](#), [citations](#)

The graphics processor (GPU) on today's commodity video cards has evolved into an extremely powerful and flexible processor. The latest graphics architectures provide tremendous memory bandwidth and computational horsepower, with fully programmable vertex and pixel processing units that support vector operations up to full IEEE floating point precision. High level languages have emerged for graphics hardware, making this computational power accessible. Architecturally, GPUs are highly parallel s ...

16 Tailoring OO analysis and design methods (panel)

 Frank Armour, Todd Cotton, Geoff Hambrick, Barbara Moo, Dennis Mancl
October 1995 **ACM SIGPLAN Notices , Proceedings of the tenth annual conference on Object-oriented programming systems, languages, and applications OOPSLA '95**, Volume 30 Issue 10

Publisher: ACM Press


Full text available:  pdf(1.68 MB) Additional Information: [full citation](#), [references](#), [index terms](#)

17 Collision detection and proximity queries

 Sunil Hadap, Dave Eberle, Pascal Volino, Ming C. Lin, Stephane Redon, Christer Ericson
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**

Publisher: ACM Press

Full text available: Additional Information:

 [pdf\(11.22 MB\)](#)[full citation](#), [abstract](#)

This course will primarily cover widely accepted and proved methodologies in collision detection. In addition more advanced or recent topics such as continuous collision detection, ADFs, and using graphics hardware will be introduced. When appropriate the methods discussed will be tied to familiar applications such as rigid body and cloth simulation, and will be compared. The course is a good overview for those developing applications in physically based modeling, VR, haptics, and robotics.

**18** Level set and PDE methods for computer graphics

David Breen, Ron Fedkiw, Ken Museth, Stanley Osher, Guillermo Sapiro, Ross Whitaker
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**

Publisher: ACM PressFull text available:  [pdf\(17.07 MB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#)

Level set methods, an important class of partial differential equation (PDE) methods, define dynamic surfaces implicitly as the level set (iso-surface) of a sampled, evolving nD function. The course begins with preparatory material that introduces the concept of using partial differential equations to solve problems in computer graphics, geometric modeling and computer vision. This will include the structure and behavior of several different types of differential equations, e.g. the level set eq ...

**19** An efficient implementation scheme of concurrent object-oriented languages on stock multicomputers

Kenjiro Taura, Satoshi Matsuoka, Akinori Yonezawa
July 1993 **ACM SIGPLAN Notices , Proceedings of the fourth ACM SIGPLAN symposium on Principles and practice of parallel programming PPOPP '93**,
Volume 28 Issue 7

Publisher: ACM PressFull text available:  [pdf\(1.18 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Several novel techniques for efficient implementation of concurrent object-oriented languages on general purpose, stock multicomputers are presented. These techniques have been developed in implementing our concurrent object-oriented language ABCL on a Fujitsu Laboratory's experimental multicomputer AP1000 consisting of 512 SPARC chips. The proposed intra-node scheduling mechanism reduces the cost of local message passing. The cost of intra-node asynchronous message passing is about 20 SPARC ...

**20** Low-level router design and its impact on supercomputer system performance

V. Puente, J. A. Gregorio, C. Izu, R. Beivide, F. Vallejo
May 1999 **Proceedings of the 13th international conference on Supercomputing ICS '99**

Publisher: ACM PressFull text available:  [pdf\(1.54 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)